

Ingeniería de Software I-II - Primer Parcial 26/9/2016

Entregar cada ejercicio en hojas separadas. Poner nombre en todas las hojas (incluida esta)

Tema 1 Ejercicio 1 - Parte Teórica (10 puntos)

Criterio: Cada ítem vale 1 punto. Se aprueba con 4 puntos.

Dadas las siguientes diez afirmaciones indique la respuesta correcta para cada una.

Por favor conteste en esta misma hoja.

1) Las clases con un único mensaje:

- A ☐ Siempre son cohesivas.
- B ☐ Reflejan un diseño incompleto.
- C ☐ Deben ser inmutables.
- D ☐ Ninguna de las anteriores.

3) En diseño siempre es deseable tener:

- A ☐ Baja cohesión y bajo acoplamiento
- B ☐ Alta cohesión y bajo acoplamiento
- C ☐ Baja cohesión y alto acoplamiento
- D ☐ Alta cohesión y alto acoplamiento

5) Como regla de buen diseño los objetos:

- A ☐ Deben ser completamente inmutables.
- B ☐ No deben tener jamás setters ni getters.
- C ☒ Su esencia debe ser inmutable.
- D ☐ Todas las de arriba.

7) Los setters son problemáticos porque:

- A ☐ Violan el encapsulamiento.
- B ☐ Podrían modificar la esencia de un objeto.
- C ☐ Podrían generar objetos inválidos.
- D ☐ Todas las de arriba.

9) Las clases abstractas mapean en la realidad con:

- A ☐ Ideas o Conceptos.
- B ☐ Herencias e Interfaces.
- C ☐ Refinamiento de Conocimiento.
- D ☐ Implicación de pre y post condiciones.

2) Dos patrones con la misma estructura:

- A ☐ Son el mismo patrón de diseño.
- B ☐ Tienen una misma intención (intent).
- C ☐ Se implementan con el mismo protocolo.
- D ☐ Ninguna de las de arriba.

4) La persistencia se debería modelar:

- A ☐ Como un método estático o de clase.
- B ☐ Implementando una Interfaz "Persistible".
- C ☐ En otro objeto.
- D ☐ En la superclase común a todos los objetos persistentes.

6) El uso de Null está desaconsejado porque:

- A ☐ No es polimórfico con otros objetos.
- B ☐ No se mapea con un ente del mundo real.
- C ☐ Se mapea en el puntero 0000 y podría dar errores de división por cero.
- D ☐ Obliga a utilizar double dispatch siempre.

8) Puede ser responsabilidad de una clase:

- A ☐ Establecer las interfaces a implementar.
- B ☐ La creación de instancias.
- C ☐ Definir métodos abstractos.
- D ☐ Ninguna de las anteriores.

10) La herencia tiene como principal objetivo:

- A ☐ Variar comportamiento en las subclases.
- B ☐ Reutilizar código.
- C ☐ Ahorrar memoria.
- D ☐ Ninguna de las anteriores.

Ingeniería de Software I-II - Primer Parcial 26/9/2016

Entregar cada ejercicio en hojas separadas. Poner nombre en todas las hojas (incluida esta)

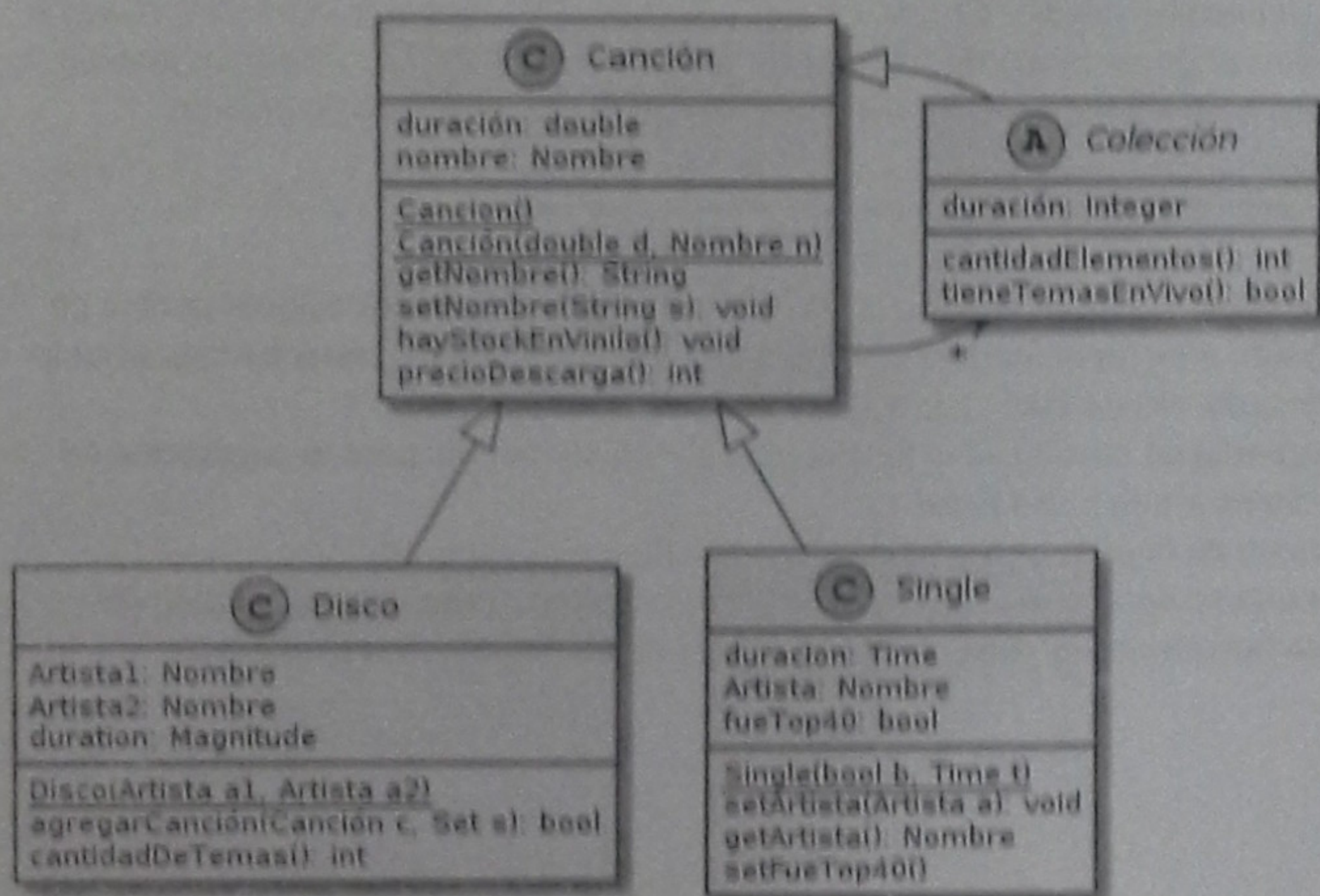
Ejercicio 2 Parte - Práctica (10 puntos)

Criterio: El ejercicio de crítica vale 4 puntos. El de diseño 6. Se aprueba con 4 puntos, con al menos 2 puntos en el ejercicio de diseño.

2.1 Crítica (4 puntos)

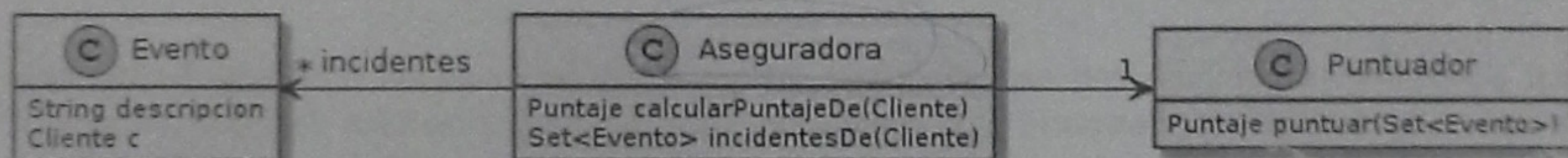
Analizar el siguiente modelo y descubrir los errores o mejoras posibles. Justificar cada observación realizada e indicar con "leve", "mediano" o "grave" la importancia de cada caso.

Nota: Se cuenta únicamente con este diagrama. Aclare cualquier asunción realizada. La correcta categorización de los errores es parte de la evaluación. Los métodos subrayados denotan métodos de clase. En el diagrama la letra C acompañando al nombre de una clase indica una clase concreta, mientras que la A una clase abstracta. Colección es entonces la única clase abstracta del diagrama.



2.2 Diseño (6 puntos)

La prestigiosa compañía TVG ofrece un seguro automotor con un sistema de puntos (scoring) que permite pagar menos a los clientes que mejor conducen. Así, los asegurados que no cometan ninguna imprudencia pagarán una cuota mínima, y este valor se irá incrementando a medida que se registren eventos inseguros como cruzar un semáforo en rojo, no respetar velocidad máxima o frenar bruscamente. De momento, el sofisticado mecanismo de detección de eventos y la traducción de puntaje a dinero quedarán por fuera del problema que nos piden atender. En cambio, quieren que nos concentremos en la asignación de puntajes a los clientes. Esencialmente, a cada cliente se le acumulan 100 puntos de penalización (pp) por cada dos eventos. Por ejemplo, se sabe que para la cliente de la compañía Marta Luisa Prost hay registro de dos eventos (uno de cruzar en rojo y una frenada brusca), por lo que se le asignan 100 pp. El siguiente diseño simplificado representa a la implementación actual:



Sin embargo, la compañía está decidida a cambiar esta política, ya que la modalidad actual no permite ponderar de distinta manera cada tipo de evento imprudente. La nueva modalidad de asignación deberá considerar los siguientes casos: "Cruzar en rojo": 200 pp, "No respetar velocidad máxima": 100 pp, "Estacionar sobre el cruce peatonal", 30 pp, "Frenada brusca": 20 pp. Con este cambio, la controvertida Marta Luisa recibiría 220 pp (200 del cruce en rojo y 20 por la frenada brusca).

Se pide: (6 puntos) (a+b+c= 5 puntos; d) 1 punto)

- Presente un diagrama de clases que modele la nueva funcionalidad para asignar puntos de penalización. Puede adaptar el modelo anterior o proponer uno nuevo. Describa brevemente el modelo presentado. Aclare cualquier asunción tomada.
- Ilustre en un diagrama de secuencia el funcionamiento de su modelo para la asignación de puntajes de la cliente Marta Luisa Prost.
- Brinde un diagrama de objetos para el escenario descrito en el punto b).
- Suponga ahora que se agrega el evento de "Aceleración brusca", con una asignación de 150 pp. Explique brevemente cómo impacta en el modelo presentado en el punto a).